CSE 06131223 ♦ CSE 06131224

Structured Programming

Lecture 5

Overview of C Language

জাতীয় কবি কাজী নজরুল ইসলাম বিশ্ববিদ্যালয়
স্থাপিত-২০০৬
ত্রিশাল, ময়মনসিংহ

Prepared by

Md. Mijanur Rahman, Prof. Dr.
Dept. of Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University, Bangladesh
www.mijanrahman.com

Contents

OVERVIEW OF C LANGUAGE

- The C Programming Language
- History of the C Language
- Features of C Language
- Preparing to C Program
- The C Development Cycle
- Your First C Program
- How C Programming Language Works?
- C Environment
- Key Uses and Applications
- Why Learn C Language?
- Sample C Programs

C Programming Language

- C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs in 1972.
- ‘C’ programming language contains all the features of the earlier languages:

  ‘ALGOL’, ‘BCPL’ and ‘B’.
- **C** is extremely popular, simple, and flexible to use.
- **C** is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more.

C Programming Language

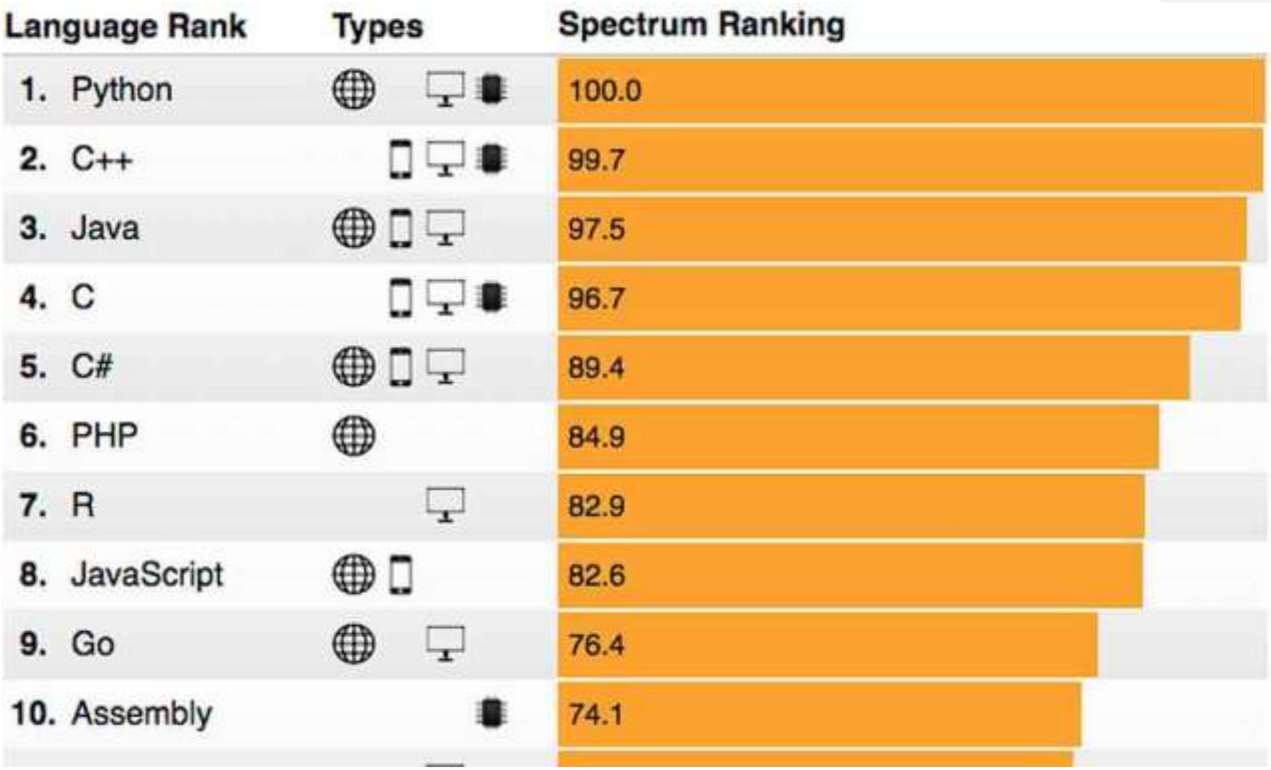
- C was invented to write an operating system called UNIX. The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C.
- Today ‘C’ runs under a variety of operating systems and hardware platforms.
- Today C is the most widely used and popular System Programming Language. Most of the state-of-the-art software have been implemented using C.
- Thus, C is a base for the programming. If you know ‘C,’ you can easily grasp the knowledge of the other programming languages that uses the concept of ‘C’.

C Programming Language

- C has now become a widely used professional language for various reasons:
  - C is base for the programming
  - High-level language, easy to learn
  - Structured language, flexible to use
  - It produces efficient programs for various applications
  - It can handle low-level activities
  - It can be compiled on a variety of computer platforms

C Programming Language

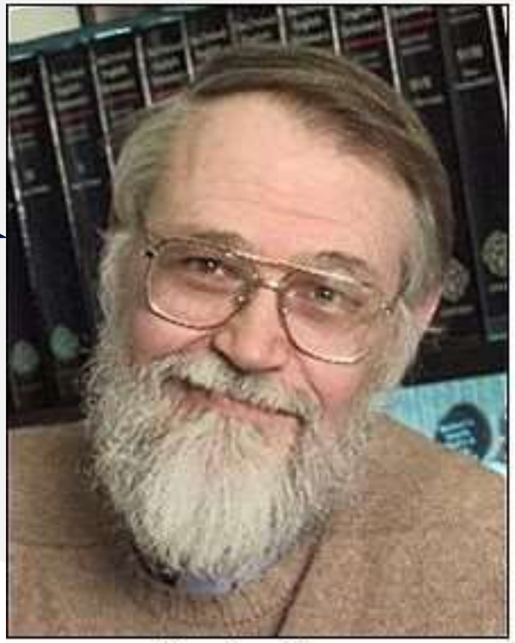
- IEEE-the best 10 top programming language in 2018



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ▮	100.0
2. C++	📱 🖥 ▮	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ▮	96.7
5. C#	🌐 📱 🖥	89.4
6. PHP	🌐	84.9
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	▮	74.1

History of C Language

- C was created by Dennis Ritchie at the Bell Telephone Laboratories in 1972.
- The C Programming Language, written by Brian Kernighan and Dennis Ritchie in 1978.
- The American National Standards Institute (ANSI) formed a committee in 1983 to establish a standard definition of C, which became known as **ANSI Standard C**.
- The C language is so named because its predecessor was called B. The B language was developed by Ken Thompson of Bell Labs.
- Dennis Ritchie and Ken Thompson also the inventors of the UNIX Operating System.



Brian Kernighan

Dennis Ritchie

Ken Thompson

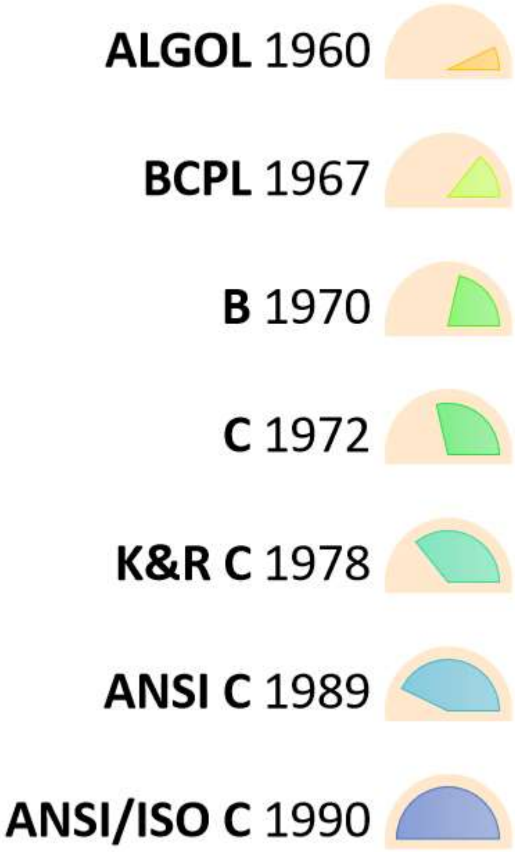
History of C Language

- **In 1960**, ‘ALGOL’ was first introduced the concept of structured programming to the developer community. **‘ALGOL, is the base or father of programming languages**.
- **In 1967**, a new computer programming language was announced called as ‘BCPL’ which stands for **Basic Combined Programming Language**. BCPL was designed and developed by Martin Richards, especially for writing system software. This was the era of programming languages.
- **In 1970**, a new programming language **called ‘B’ was introduced by Ken Thompson that contained multiple features of ‘BCPL.’** This programming language was created using UNIX operating system at AT&T and Bell Laboratories. Both the ‘BCPL’ and ‘B’ were system programming languages.

History of C Language

- **In 1972**, a great computer scientist Dennis Ritchie created a new programming language called ‘C’ at the Bell Laboratories. It was created from ‘ALGOL’, ‘BCPL’ and ‘B’ programming languages. ‘C’ programming language contains all the features of these languages.
- Initially ‘C’ programming was limited to the UNIX operating system, but today ‘C’ runs under a variety of operating systems and hardware platforms.
- **In 1989**, the American National Standards Institute (ANSI) defined a commercial standard for ‘C’ language. Later, it was approved by the International Standards Organization (ISO) in 1990.
- Thus, ‘C’ programming language is also called as ‘ANSI C’.

History of C Language



ALGOL 1960

BCPL 1967

B 1970

C 1972

K&R C 1978

ANSI C 1989

ANSI/ISO C 1990

- Languages, such as C++/Java are developed from ‘C’. These languages are widely used in various technologies.
- Thus, ‘C’ forms a base for many other languages that are currently in use.

C and the ANSI Standard

- **The American National Standards Institute (ANSI) to build a standard for the C language in 1983.** ANSI approved the application and formed the X3J11 Technical Committee to work on the C standard.
- **By the end of 1989, the committee approved the ANSI standard for the C programming language.**
- The ANSI standard for C enhances the K&R standard and defines a group of commonly used C functions that are expected to be found in the ANSI C standard library.
- Now, all C compilers have the standard library, along with some other compiler-specific functions.

Major Features of C Language

- There are several reasons why many computer professionals feel that C is at the top of the list:
  - **C is a powerful and flexible language.** C is used for projects as diverse as operating systems, word processors, graphics, spreadsheets, and even compilers for other languages.
  - **C is a popular language preferred by professional programmers.** As a result, a wide variety of C compilers and helpful accessories are available.
  - **C is a portable language.** *Portable* means that a C program written for one computer system (an IBM PC, for example) can be compiled and run on another system (a DEC VAX system, perhaps) with little or no modification. Portability is enhanced by the ANSI standard for C, the set of rules for C compilers.
  - **C is a language of few words, containing only a handful of terms, called *keywords,* which serve as the base on which the language's functionality is built.** As you program with C, you will find that it can be programmed to do any task.
  - **C is modular. C code can (and should) be written in routines called *functions*.** These functions can be reused in other applications or programs. By passing pieces of information to the functions, you can create useful, reusable code.

Major Features of C Language

- High-level programming languages, including C, have the following advantages:
  - **Readability:** Programs are easy to read.
  - **Maintainability:** Programs are easy to maintain.
  - **Portability:** Programs are easy to port across different computer platforms.
- The C language's readability and maintainability benefit directly from its relative closeness to human languages, especially English.

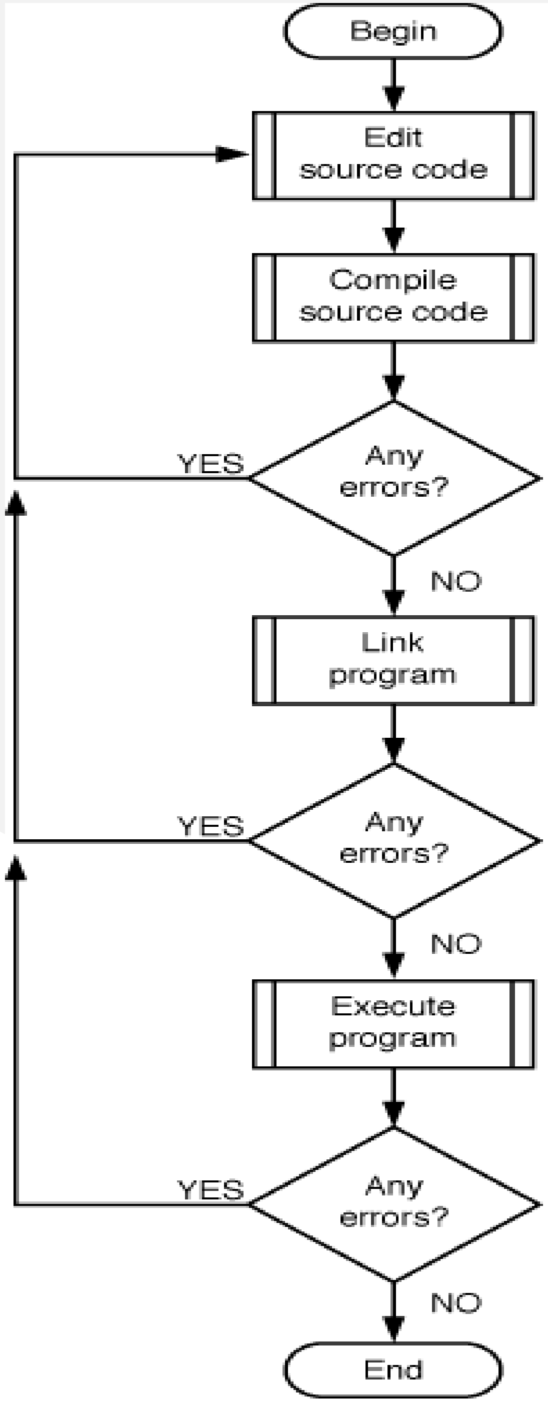
Preparing to C Program

- When creating a program in C (or for that matter, a computer program in any language), you should follow a similar sequence of steps:
  1. Determine the objective(s) of the program.
  2. Determine the methods you want to use in writing the program.
  3. Create the program to solve the problem.
  4. Run the program to see the results.

The C Development Cycle

- **Step 1: Editing**
  - Use an editor to write your source code. By tradition, C source code files have the extension .C (for example, MYPROG.C, DATABASE.C, and so on).
- **Step 2: Compiling**
  - Compile the program using a compiler. If the compiler doesn't find any errors in the program, it produces an object file. The compiler produces object files with an .OBJ extension and the same name as the source code file (for example, MYPROG.C compiles to MYPROG.OBJ).
  - If the compiler finds errors, it reports them. You must return to step 1 to make corrections in your source code.

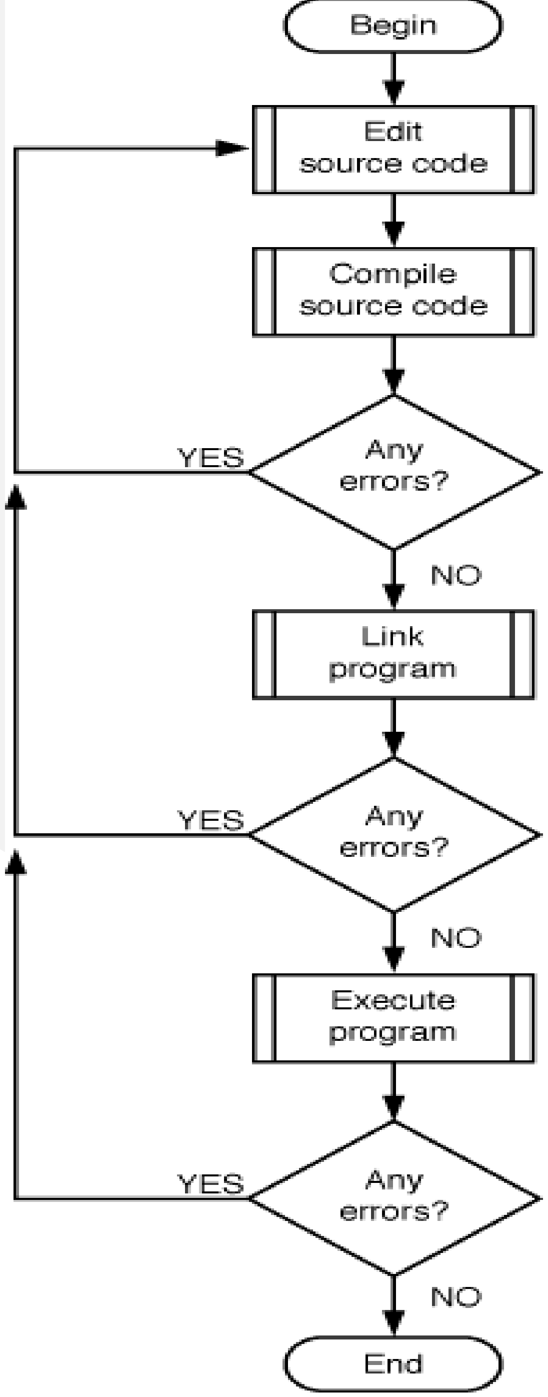
Fig: *The steps involved in C program development.*



The C Development Cycle

- **Step 3: Linking**
  - Link the program using a linker. If no errors occur, the linker produces an executable program located in a disk file with an .EXE extension and the same name as the object file (for example, MYPROG.OBJ is linked to create MYPROG.EXE).
- **Step 4: Executing**
  - Execute the program. You should test to determine whether it functions properly. If not, start again with step 1 and make modifications and additions to your source code.

Fig: *The steps involved in C program development.*



Your First C Program

- A **C program** can vary from 3 lines to millions of lines and it should be written into one or more text files with extension **".c"**; for example, *hello.c*. This demonstration uses a program named HELLO.C, which does nothing more than display the words **Hello, World!** on-screen.
- The source code for HELLO.C :

```
/* This is my first C program */
#include <stdio.h>
#include <conio.h>
int main()
{
    printf("Hello, World!\n");
    return 0;
    getch();
}
```

Your First C Program

- Once your compiler and editor are ready, follow these steps to enter, compile, and execute HELLO.C.
- **Entering and Compiling HELLO.C**
  - You have just entered, compiled, and run your first C program. Admittedly, HELLO.C is a simple program that doesn't do anything useful, but it's a start.
- **Comments**
  - The first line contains a comment:

```
/* This is my first C program */
```

- You notice that this line starts with a combination of a slash and an asterisk, /*, and ends with */. In C, /* is called the opening comment mark, and */ is the closing comment mark. The C compiler ignores everything between the opening comment mark and closing comment mark. Most C compilers allow you to write a comment that crosses more than one line. For instance, you can write a comment in C like this:

```
/* This comment does not increase the size of   the executable file (binary code), nor does   it affect the performance speed.*/
```

- You can also use double slash // for single line comment, like:

```
// This is my first C program
```

Your First C Program

- **The #include Directive**

  ```
  #include <stdio.h>
  ```

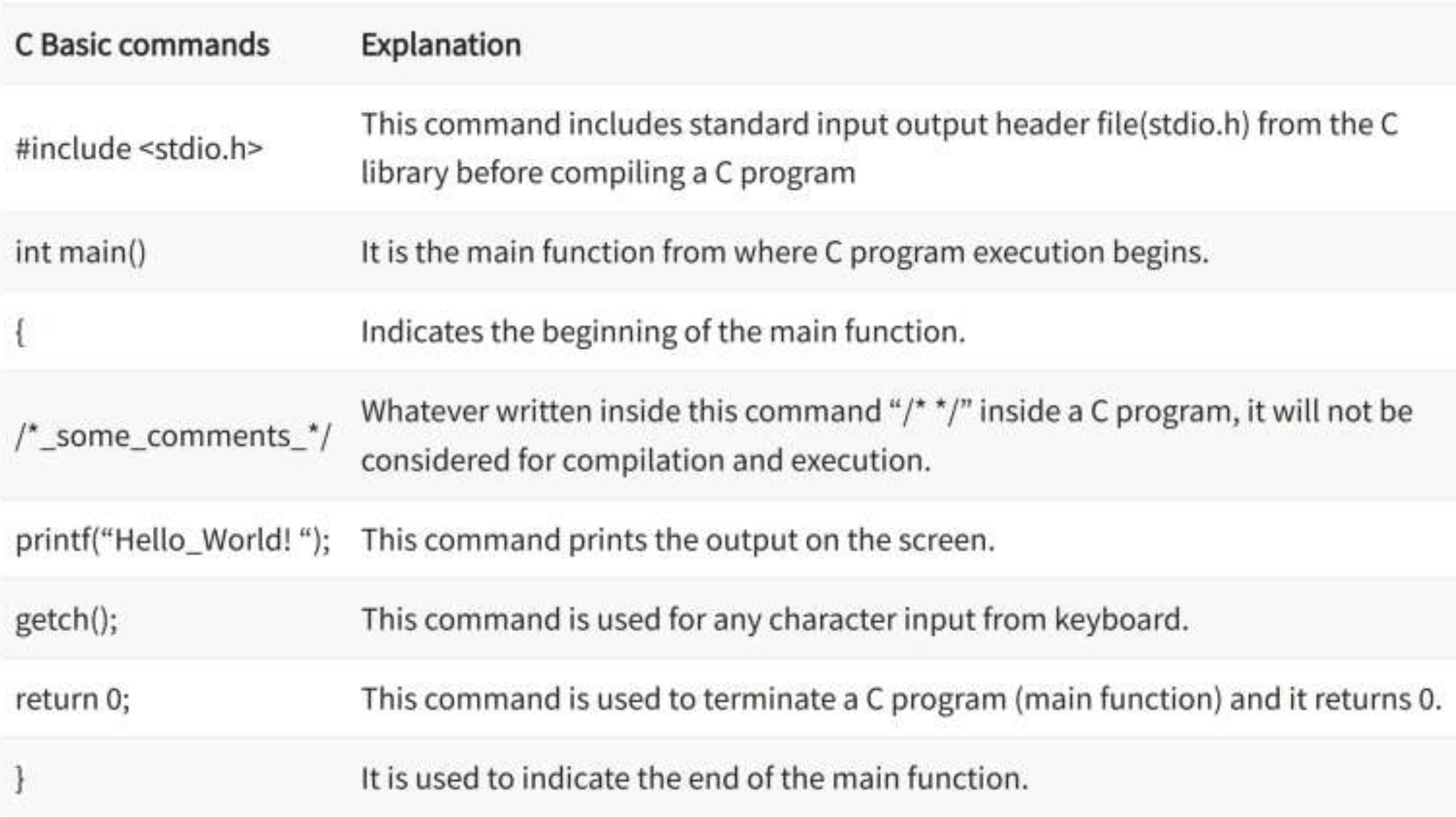
  - You see that this line starts with a pound sign, #, which is followed by include. In C, #include forms a preprocessor directive that tells the C preprocessor to look for a file and place the contents of the file in the location where the #include directive indicates.
  - The preprocessor is a program that does some preparations for the C compiler before your code is compiled.
- **Header Files**
  - The files that are required by the #include directive, like stdio.h, are called header files because the #include directives are almost always placed at the head of C programs. Actually, the extension name of .h does mean "header." Besides stdio.h, there are more header files, such as stdlib.h, string.h, math.h, and so on.

Your First C Program

- **The main() Function**
    - This is a very special function in C. Every C program must have a main() function, and every C program can only have one main() function. You can put the main() function wherever you want in your C program. However, the execution of your program always starts with the main() function.
- **The Newline Character (\n)**
    - In the printf() function, one thing worth mentioning at this moment is the newline character, \n. Usually suffixed at the end of a message, the newline character tells the computer to generate a carriage-return and line-feed sequence so that anything printed out after the message will start on the next new line on the screen.
- **The return Statement**
    - All functions in C can return values. For instance, when you make a function to add two numbers, you can make such a function that returns to you the value of the addition.
    - The main() function itself returns a value. By default, main() returns an integer. In C, integers are decimal numbers without fraction portions.

Your First C Program

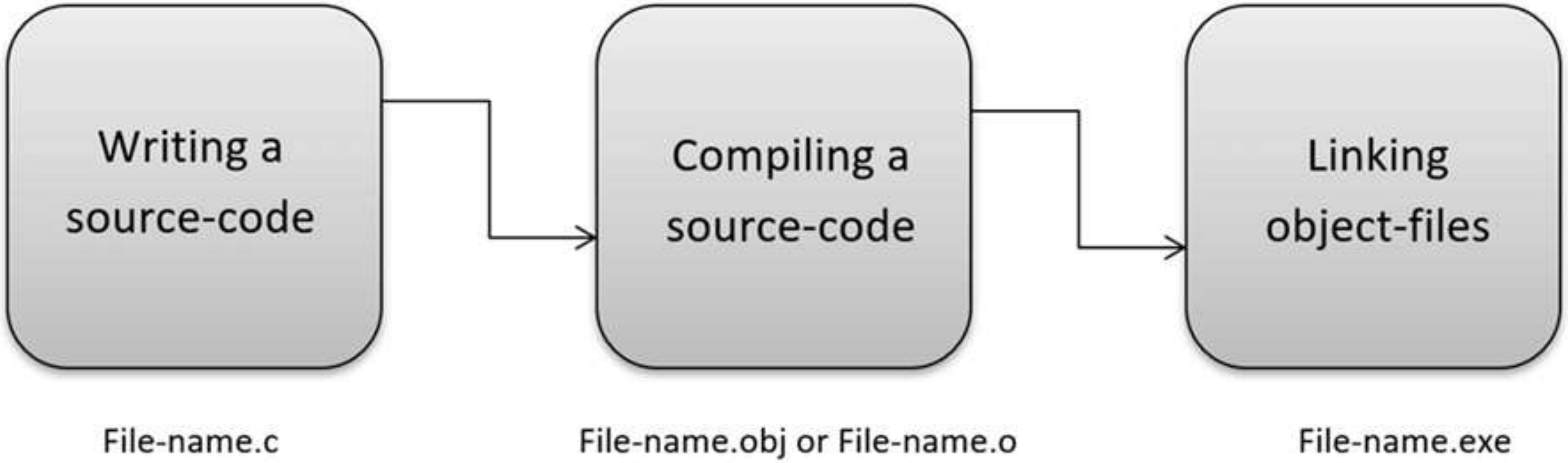
- Following are the basic commands in C programming language:



C Basic commands	Explanation
#include <stdio.h>	This command includes standard input output header file(stdio.h) from the C library before compiling a C program
int main()	It is the main function from where C program execution begins.
{	Indicates the beginning of the main function.
/*_some_comments_*/	Whatever written inside this command "/* */" inside a C program, it will not be considered for compilation and execution.
printf(“Hello_World! “);	This command prints the output on the screen.
getch();	This command is used for any character input from keyboard.
return 0;	This command is used to terminate a C program (main function) and it returns 0.
}	It is used to indicate the end of the main function.

How C Programming Language Works?

- C is a compiled language. **A compiler is a special tool that compiles the program and converts it into the object file which is machine readable.**
- After the compilation process, **the linker will combine different object files and creates a single executable file to run the program.**
- The following diagram shows the execution of a ‘C’ program.



Writing a source-code

Compiling a source-code

Linking object-files

File-name.c

File-name.obj or File-name.o

File-name.exe

C Environment

- If you want to set up your environment for C programming language, you need the following two software tools available on your computer:
    1. **Text Editor, and**
    2. **The C Compiler**
- **Text Editor**
    - This will be used to type your program. Examples of few a editors include Windows Notepad, Notepad++, Netbeans, Code::Blocks, Eclipse CDT, CodeLite IDE, Microsoft’s Visual Studio Code Editor, and vim or vi.
    - The name and version of text editors can vary on different operating systems. For example, Notepad will be used on Windows, and vim or vi can be used on windows as well as on Linux or UNIX.
    - The files you create with your editor are called the source files and they contain the program source codes. The source files for C programs are typically named with the extension “***.c**".

C Environment

- **The C Compiler**
  - The source code written in source file is the human readable source for your program. It needs to be "compiled", into machine language so that your CPU can actually execute the program as per the instructions given.
  - The compiler compiles the source codes into final executable programs. The most frequently used and free available compiler is the GNU C/C++ compiler (GCC). Some other C compilers include: Turbo C, Borland Turbo C, Clang, Microsoft’s C++ compiler, Visual C++, etc.
- **Codeblocks** is open-source IDE for writing programs in C. It supports GCC (GNU Compiler Collection).

C Environment

- The C compilation system consists of a compiler, an assembler, and a link editor. The cc command invokes each of these components automatically unless you use command-line options to specify otherwise.
- The following figure shows the organization of the C compilation system:

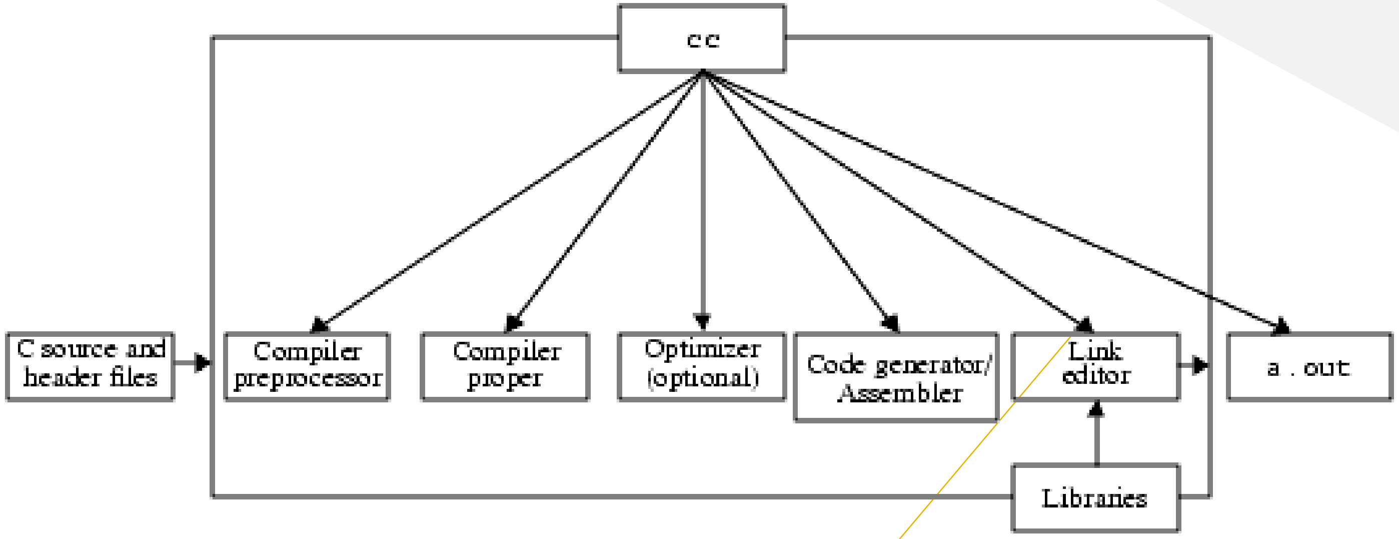


FIG: Organization of the C Compilation System

Key Uses and Applications

- **Where is C used?**
  - ‘C’ language is widely used in embedded systems.
  - It is used for developing system applications.
  - It is widely used for developing desktop applications.
  - Most of the applications by Adobe are developed using ‘C’ programming language.
  - It is used for developing browsers and their extensions.
  - It is used to develop databases. MySQL is the most popular database software which is built using ‘C’.
  - It is used in developing an operating system. Operating systems such as Apple’s OS X, Microsoft’s Windows, Unix/Linux, and Symbian are developed using ‘C’ language. It is used for developing desktop as well as mobile phone’s operating system.
  - It is used for compiler production.
  - It is widely used in IOT applications.

Why learn C Language?

- As ‘C’ is a base language for many programming languages, so, learning ‘C’ as the main language will play an important role while studying other programming languages.
- ‘C’ can be used widely in various applications. It is a simple language and provides faster execution. There are many jobs available for a ‘C’ developer in the current market.
- ‘C’ is a structured programming language in which program is divided into various modules. Each module can be written separately and together it forms a single ‘C’ program. This structure makes it easy for testing, maintaining and debugging processes.
- ‘C’ contains 32 keywords, various data types and a set of powerful built-in functions that make programming very efficient.
- Another feature of ‘C’ programming is that it can extend itself. A ‘C’ program contains various functions which are part of a library. We can add our features and functions to the library. We can access and use these functions anytime we want in our program. This feature makes it simple while working with complex programming.
- Various compilers are available in the market that can be used for executing programs written in this language.
- It is a highly portable language which means programs written in ‘C’ language can run on other machines. This feature is essential if we wish to use or execute the code on another computer.

Sample Programs

- **Adding Two Numbers:**

```
/* Write a C program to add two numbers. */
#include <stdio.h>
int main()
{
          int a = 10;
          int b = 20;
          int c = a + b;
          printf("Sum of two numbers: %d",c);
          return 0;
}
```

Output:

```
Sum of two numbers: 30
```

Sample Programs

- **Adding Two Numbers:**

```
/* Write a C program to add two numbers. */
#include <stdio.h>
int main()
{
        int a, b, c;
        printf(“Enter two numbers: \n”);
        scanf(“%d %d”, &a, &b);
        c = a + b;
        printf("Sum of two numbers: %d",c);
        return 0;
}
```

Output:

```
Enter two numbers:
20 30
Sum of two numbers: 50
```

Sample Programs

- **Calculating the area of a circle:**

```c
/* Write a program to calculate the area of a circle */
#include <stdio.h>
int main()
{
    float rad, area;
    printf(“Enter the radius of a circle: \n”);
    scanf(“%f”, &rad);
    area = 3.14 * rad * rad;
    printf(“The area is: %.2f", area);
    return 0;
}
```

Output:

```
Enter the radius of a circle:
5.0
The area is: 78.05
```

Sample Programs

- **Print ASCII Value:**

```c
/* Program to Print ASCII Value */
#include <stdio.h>
int main()
{
  char c;
  printf("Enter a character: ");
  scanf("%c", &c);
  printf("ASCII value of %c = %d", c, c);
  // %d displays the integer value
  // %c displays the actual character
  return 0;
}
```

Output:

```
Enter a character: A
ASCII value of A = 65
```

?

THE END